

Amendments to the claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of claims:

1. (currently amended) A method for providing dynamic interaction between a pair of application programs by an ~~platform-neutral~~ interface module of a terminal, the pair of applications including a requestor application desiring access to a target application, the method comprising the steps of:

registering access information of the target application, the access information including published access information made available in a data structure for retrieval by the ~~platform-neutral~~ interface module;

receiving an access request by the ~~platform-neutral~~ interface module from the requestor application, the access request including request content corresponding to the published access information of the target application;

obtaining an interface component by using the request content to search the data structure, the interface component configured for enabling communication between the ~~platform-neutral~~ interface module and the target application in an access format expected by the target application; and

employing the interface component by the ~~platform-neutral~~ interface module to satisfy the access request of the requestor application for interaction with the target application.

2. (currently amended) The method according to claim 1, wherein the target application is selected from the group comprising: the target application configured for communication in a compatible language with the platform-neutral interface; and the target application interface component includes an application

program interface configured for communication in a ~~incompatible~~ language incompatible with the ~~platform-neutral~~ interface module.

3. (original) The method according to claim 2, wherein the incompatible language is that used by a native runtime environment of the terminal.
4. (currently amended) The method according to claim ~~[[2]]~~ 1, wherein the interface component is includes an application program interface (API) ~~expressed in the~~ configured for communication in a language compatible with the interface module language.
5. (currently amended) The method according to claim 2, wherein the interface component further includes is an access handler ~~extension element~~ configured for providing ~~mediation~~ translation between the ~~platform-neutral~~ interface module and the application program interface ~~target application~~ expressed in the incompatible language.
6. (currently amended) The method according to claim 5 further comprising the step of registering the access handler ~~extension element~~ with the ~~platform-neutral~~ interface module through an extension interface, the published access information of the access handler ~~extension element~~ being added to the data structure.
7. (currently amended) The method according to claim 6 further comprising the step of accessing the target application through the ~~platform-neutral~~ interface module using the access handler ~~extension element~~ to call a corresponding application program interface (API) ~~expressed in the incompatible language of the target application~~.
8. (currently amended) The method according to claim ~~[[2]]~~ 7 further comprising the step of employing a search algorithm with the request content for identifying matching ones of the access handlers ~~interface component~~ for use by the ~~platform-neutral~~ interface module.

9. (currently amended) The method according to claim 8, wherein the language used to express the ~~platform-neutral~~ interface module is selected from the group comprising; a structured definition language and a script.
10. (original) The method according to claim 9, wherein the structured definition language is based on XML.
11. (original) The method according to claim 9, wherein the language used to express the script is ECMAScript.
12. (currently amended) The method according to claim ~~[[2]]~~ 1 further comprising the a further step of assembling the request content to include selected from the group comprising; a local location and a remote location.
13. (original) The method according to claim 12, wherein the remote location is on another terminal coupled to said terminal through a network, the other terminal having one of the pair of applications for network interaction with the other of the pair of applications.
14. (currently amended) The method according to claim 13, wherein said other terminal is configured as a client of a schema defined service accessible over the network.
15. (original) The method according to claim ~~[[2]]~~ 1, wherein the data structure is selected from the group comprising an application profile table and an application API descriptor table.
16. (original) The method according to claim 15, wherein the application profile table includes application profiles of a plurality of target applications.
17. (currently amended) The method according to claim 15, wherein the application API descriptor table includes descriptors selected from the group comprising; API descriptors and ~~extension-element~~ access handler descriptors.

18. (original) The method according to claim 15, wherein the data structure includes the access information selected from the group comprising; application URI, application version, application description, and a predefined set of matching API construct pairs.

19. (currently amended) The method according to claim ~~[[2]]~~ 7 comprising a further ~~comprising the step of providing an interface of the platform neutral~~ interface module selected from the group comprising; an extension interface, a query and registration interface, and an execution interface.

20. (currently amended) The method according to claim 19, wherein the extension interface is configured for dynamically extending a coupling of a new said interface component to the ~~platform neutral~~ interface module.

21. (currently amended) A terminal for providing dynamic interaction between a pair of application programs in a platform neutral environment provided by the runtime environment of the terminal, the pair of applications including a requestor application desiring access to a target application, the terminal comprising memory and a computer processor configured to implement instructions stored on the memory, the instructions configured to provide:

a data structure for registering access information of the target application, the access information including published access information;

an interface module for providing the platform neutral environment, the interface module configured for receiving an access request from the requestor application, the access request configured to include request content corresponding to the published access information of the target application, the published access information of the data structure retrievable by the interface module; and

an interface component coupled to the interface module retrievable by using the request content to search the data structure, the interface component

configured for enabling communication between the interface module and the target application in an access format expected by the target application;

wherein employing the interface component by the interface module satisfies the access request of the requestor application in interaction with the target application.

22. (currently amended) The terminal according to claim 21, wherein the ~~target application is selected from the group comprising: the target application~~ configured for communication in a compatible language with the interface module; and the target application interface component includes an application program interface configured for communication in a ~~compatible~~ language incompatible with the interface module.

23. (original) The terminal according to claim 22, wherein the incompatible language is that used by the native runtime environment of the terminal.

24. (currently amended) The terminal according to claim ~~[[22]]~~ 21, wherein the interface component ~~is~~ includes an application program interface (API) configured for communication in a language expressed in the compatible language with the interface module.

25. (currently amended) The terminal according to claim 22, wherein the interface component further includes is an extension element access handler configured for providing ~~mediation~~ translation between the interface module and the application program interface ~~target application expressed in the incompatible language.~~

26. (currently amended) The terminal according to claim 25 further comprising an extension interface for registering the ~~extension element~~ access handler with the interface module, the published access information of the ~~extension element~~ access handler being added ~~configured for adding~~ to the data structure.

27. (currently amended) The terminal according to claim 26 further comprising a corresponding application program interface (API) callable by the ~~extension element~~ access handler ~~step~~ for accessing the target application through the interface module, ~~the application program interface (API) expressed in the incompatible language of the target application.~~

28. (currently amended) The terminal according to claim ~~[[22]]~~ 27 further comprising a search algorithm for using the request content to identify matching ones of the ~~extension element~~ access handlers for use by the interface module.

29. (original) The terminal according to claim 28, wherein the language used to express the interface module is selected from the group comprising; a structured definition language and a script.

30. (original) The terminal according to claim 29, wherein the structured definition language is based on XML.

31. (original) The terminal according to claim 29, wherein the language used to express the script is ECMAScript.

32. (currently amended) The terminal according to claim ~~[[22]]~~ 27, wherein the request content is configured to include selected from the group comprising; a local location and a remote location.

33. (original) The terminal according to claim 32, wherein the remote location is on another terminal coupled to said terminal through a network, the other terminal having one of the pair of applications for network interaction with the other of the pair of applications.

34. (currently amended) The terminal according to claim 33, wherein said other terminal is configured as a client of a schema defined service accessible over the network.

35. (currently amended) The terminal according to claim ~~[[22]]~~ 27, wherein the data structure is selected from the group comprising; an application profile table and an application API descriptor table.

36. (original) The terminal according to claim 35, wherein the application profile table includes application profiles of a plurality of target applications.

37. (currently amended) The terminal according to claim 35, wherein the application API descriptor table includes descriptors selected from the group comprising; API descriptors and ~~extension element~~ access handler descriptors.

38. (original) The terminal according to claim 35, wherein the data structure includes the access information selected from the group comprising; application URI, application version, application description, and a predefined set of matching API construct pairs.

39. (original) The terminal according to claim 22 further comprising an interface of the interface module selected from the group comprising; an extension interface, a query and registration interface, and an execution interface.

40. (original) The terminal according to claim 39, wherein the extension interface is configured for dynamically extending a coupling of a new said interface component to the interface module.

41. (original) The terminal according to claim 39, wherein the query and registration interface is configured for publishing the access information related to the interface component.

42. ~~A computer program product~~ memory comprising instructions for providing dynamic interaction between a pair of application programs in a platform neutral environment provided by a runtime environment of a terminal, the pair of applications including a requestor application desiring access to a target

application, the ~~computer program product~~ memory comprising instructions for execution by a processor to implement:

~~— a computer readable medium;~~

a data structure module ~~stored on the computer readable medium~~ for registering access information of the target application, the access information including published access information;

an interface module coupled to the data structure module for providing the platform neutral environment, the interface module configured for receiving an access request from the requestor application, the access request configured to include request content corresponding to the published access information of the target application, the published access information of the data structure module retrievable by the interface module; and

an interface component module coupled to the interface module, the interface module configured for containing an interface element retrievable by using the request content to search the data structure module, the interface component configured for enabling communication between the interface module and the target application in an access format expected by the target application;

wherein employing the interface component by the interface module satisfies the access request of the requestor application in interaction with the target application.